

## Занятие 2: Лексика языка и типы данных



Кодировка: UTF-16

## **Виды лексем:**

- ✓ Пробелы (пробел, табуляция, или перевод строки)
- ✓ Комментарии
  - строчный //
  - блочный /\* \*/
  - JavaDoc /\*\* \* \*/ )
- ✓ Сами лексемы

- ✓ Идентификаторы
  - они должны начинаться с буквы или символа подчеркивания;
  - они могут содержать латинские буквы, символы подчеркивания или цифры без пробелов;
  - названия идентификаторов не должны совпадать с ключевыми словами;
- ✓ Ключевые слова
- ✓ Литералы
  - целочисленные (5)
  - дробные (1.3, .5)
  - логические (true-false)
  - символьные ('a')
  - строковые ("Это строка /n а это экранированный символ")
  - null
  - разделители ( ) { } [ ] ; . ... @ ::
- ✓ Операторы (арифметические и логические)

# Ключевые слова Java

abstract  
assert  
boolean  
break  
byte  
case  
catch  
char  
class  
const  
continue  
default

double  
do  
else  
enum  
extends  
final  
finally  
float  
for  
goto  
if  
implements

import  
instanceof  
int  
interface  
long  
native  
new  
package  
private  
protected  
public  
return

short  
static  
strictfp  
super  
switch  
synchronized  
this  
throw  
throws  
transient  
try  
void  
volatile  
while

# Java code conventions

Camel case - стиль записи идентификаторов, при котором каждое слово начинается с большой буквы.

Примеры:

- ✓ MyFirstClass
- ✓ myFirstVariable
- ✓ DateToGregorianCalendarAdapter

Бывает Upper camel case и Lower camel case. Отличия в регистре первой буквы.

В java приняты следующие договоренности:

- ✓ Имена классов пишутся в Upper
- ✓ Имена экземпляров классов, методов и полей пишут в Lower

# Переменные Java

Состоят из Имени, типа и значения.

```
int i = 5;
```

```
String myString = "My new String variable";
```

В ЖК переменных есть два этапа: объявление и инициализация. Во время объявления указывается тип и имя. Во время инициализации задается начальное значение. Инициализацию и объявление можно разнести во времени:

```
String myString;
```

```
myString = "My new String variable";
```

# Примитивные типы данных

Type	Description	Default	Size	Example Literals
boolean	true or false	false	1 bit	true, false
byte	twos complement integer	0	8 bits	(none)
char	Unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\\', '\", '\n', '\b'
short	twos complement integer	0	16 bits	(none)
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d

# Типы-обертки

Каждый примитивный тип в Java имеет свой аналог в виде объекта. Так называемые Обертки (wrappers).

Обертки нужны для того, чтобы использовать примитивы как объекты. Например, хранить их в коллекциях.

Над обертками упрощены операции приведения типов.

int - Integer

long - Long

double - Double

и т. д.



**Строка** — объект, состоящий из последовательности символов.

Для создания и манипулирования строками Java платформа предоставляет общедоступный финальный (не может иметь подклассов) класс `java.lang.String`.

Данный класс является неизменяемым (*immutable*) — созданный объект класса `String` не может быть изменен. При попытке изменения строки в качестве результата будет возвращена новая строка. Если ссылок на старую переменную не осталось, она будет собрана Garbage Collector-ом

# Ссылочные типы

`.` - оператор обращения к полям и методам (`myObject.field`)

**instanceOf** - проверка принадлежности объекта классу

`==` - сравнение (надо отличать от `=`, присвоение). Прimitives сравниваются по значению. Ссылочные сравниваются по ссылке.

**()** - приведение типов

Тернарный оператор (**`a ? b : c`**)

`+` -конкатенация.

# Содержимое объекта

Система	32 бит	64 бит
Размер заголовка	8 байт	16 байт

- ✓ Заголовок объекта;
- ✓ Память для примитивных типов;
- ✓ Память для ссылочных типов;
- ✓ Смещение/выравнивание

Структура заголовка объекта:

- ✓ Hash Code
- ✓ Garbage Collection Information
- ✓ Lock
- ✓ Type Information Block Pointer
- ✓ Array Length

# Методы класса Object

- ✓ `public final native Class getClass()`
- ✓ `public native int hashCode()`
- ✓ `public boolean equals(Object obj)`
- ✓ `protected native Object clone() throws CloneNotSupportedException`
- ✓ `public String toString()`
- ✓ `public final native void notify()`
- ✓ `public final native void notifyAll()`
- ✓ `public final native void wait(long timeout) throws InterruptedException`
- ✓ `public final void wait(long timeout, int nanos) throws InterruptedException`
- ✓ `public final void wait() throws InterruptedException`
- ✓ `protected void finalize() throws Throwable`

# Инициализация классов

- ✓ Статические поля класса Parent;
- ✓ Статический блок инициализации класса Parent
- ✓ Статические поля класса Child;
- ✓ Статический блок инициализации класса Child
- ✓ Нестатические поля класса Parent
- ✓ Нестатический блок инициализации класса Parent
- ✓ Конструктор класса Parent
- ✓ Нестатические поля класса Child
- ✓ Нестатический блок инициализации класса Child
- ✓ Конструктор класса Child